Joachim Rang

An analysis of the Prothero–Robinson example for constructing new adaptive ESDIRK methods of order 3 and 4

Informatikbericht Nr. 2014-06

Institute of Scientific Computing
Carl-Friedrich-Gauss-Fakultät
Technische Universität Braunschweig

Braunschweig, Germany

This document was created April 2014 using $\LaTeX\,2_\varepsilon$.

Institute of Scientific Computing
Technische Universität Braunschweig
Hans-Sommer-Straße 65
D-38106 Braunschweig, Germany

url: www.wire.tu-bs.de
mail: wire@tu-bs.de

An analysis of the Prothero–Robinson example for constructing new adaptive ESDIRK methods of order 3 and 4

Joachim Rang

Institute of Scientific Computing, TU Braunschweig
wire@tu-bs.de

Abstract

Diagonally implicit Runge-Kutta methods with an explicit first stage (ESDIRK–methods) have usually order reduction if they are applied on stiff ODEs such as the example of Prothero and Robinson. It can be observed that the numerical order of convergence drops down to the stage order, which is limited by two. In this paper we analyse the Prothero-Robinson example and derive new order conditions to avoid order reduction. New third and fourth order ESDIRK–methods are created. The new schemes are applied on the Prothero–Robinson exampl and, on an index-2 DAE. Numerical examples show that the new methods have better convergence properties than usual ESDIRK methods.

Keywords: ODEs, order reduction, DIRK methods, adaptivity

1 Introduction

One possibility to solve stiff ODEs as the example of Prothero and Robinson [14] or differential algebraic equations are Runge-Kutta methods [7, 18]. Explicit methods may not be a good choice since for getting a stable numerical solution a stepsize restriction should be accepted, i.e. the problem should be solved with very small timesteps. Therefore it might be better to use implicit methods, as for example implicit Runge-Kutta methods, or linear-implicit methods such as Rosenbrock–Wanner methods. But in these cases the convergence may not be achieved [7, 18], i.e. the so-called order reduction phenomenom can be observed. In [7] convergence results for implicit Runge–Kutta methods applied on the example of Prothero and Robinson [14] can be found where the so-called stage order plays an important role. Ostermann and Roche prove in [13] that implicit Runge–Kutta methods may have a fractional order of convergence for general linear ODEs.

Fully implicit Runge–Kutta methods may be ineffective for solving high dimensional ODEs since they need a high computational effort which can be reduced if diagonally implicit Runge–Kutta (DIRK) methods are used. We distinguish two classes of DIRK methods. First we have the singly-diagonally-implicit Runge–Kutta (SDIRK) methods, where all diagonal elements are non-zero and equal. In this case the stage order is limited to one. Therefore in [2] Cameron introduced the so-called quasi stage order. This concept is improved in paper [3] where the method SDIRK2 is derived. In [15] an analysis of the example of Prothero and Robinson applied on SDIRK methods can be found. Numerical examples in this note show that third order is reached.

Stage order 2 is possible if the first diagonal entry is equal to zero. In this case the methods are called "explicit singly-diagonally-implicit Runge–Kutta (ESDIRK) methods". These methods are widely used in the solution of ODEs and PDEs [10, 9]. The order reduction can be decreased if order conditions for index-2 DAEs [6, 8] are satisfied, as is shown in [19, 17].

The main task of this paper is the generalisation of the concept in [15] to ESDIRK methods to develop further order conditions. With the help of these conditions we create more effective ESDIRK methods in this paper.

The paper is sturctured as follows. We first consider diagonally implicit Runge–Kutta methods with an explicit first stage and apply them to the Prothero–Robinson example. In Section 3 we are considering the local error of these methods in the stiff case. We will see that we get further order conditions which are needed to decrease the order reduction. Two third order and one fourth order ESDIRK methods are created in Section 4, and finally we present some numerical results and apply our new methods on several test examples. In the case of the Prothero–Robinson we show that we reach full order p. Since our new methods satisfy order conditions for DAEs of index 2 our second example is index-2 DAE. It is shown that our new methods are much more effective than the known ones.

2 ESDIRK–methods

2.1 Application to ODEs

We start our considerations with an ODE of the form

$$\dot{\mathbf{u}} = \mathbf{F}(t, \mathbf{u}), \quad \mathbf{u}(0) = \mathbf{u}_0. \tag{1}$$

A Runge–Kutta method (RK method) with s internal stages [7, 18] is a one–step–method for solving (1) of the form

$$\mathbf{k}_i = \mathbf{F}\left(t_m + c_i \tau_m, \mathbf{U}_i\right), \quad \mathbf{U}_i = \mathbf{u}_m + \tau_m \sum_{j=1}^{s} a_{ij} \mathbf{k}_j, \quad i = 1, \ldots, s, \tag{2}$$

$$\mathbf{u}_{m+1} = \mathbf{u}_m + \tau_m \sum_{i=1}^{s} b_i \mathbf{k}_i. \tag{3}$$

The coefficients a_{ij}, b_i and c_i should be chosen in such a way that some order conditions are satisfied to obtain a sufficient consistency order. In this paper the coefficients of the RK–method (2)–(3) satisfy $a_{ij} = 0$ for $i < j$, $i, j \in \{1, \ldots, s\}$, $a_{11} = 0$, and $a_{ii} \neq 0$ for $i \in \{2, \ldots, s\}$. RK–methods satisfying these conditions are called "explicit singly–diagonally–implicit Runge–Kutta methods (ESDIRK–methods)". These methods are discussed in several papers and books, e.g. in [18, 7]. Butcher introduces in [1] the so-called simplifying conditions, which are given by

$$\begin{aligned}
B(p): \quad & \sum_{i=1}^{s} b_i c_i^{k-1} &&= 1/k, && k = 1, \ldots, p, \\
C(q): \quad & \sum_{j=1}^{s} a_{ij} c_j^{k-1} &&= c_i^k / k, && i = 1, \ldots, s, k = 1, \ldots, q.
\end{aligned}$$

Solving a stiff ODE with the help of a RK–method, the convergence order may drop down from p to q, if $p > q$, see [7], i.e. the method has order reduction. The minimum of p and q is often called stage order of the Runge-Kutta-method.

Runge–Kutta methods have the advantage that they allow an easy implementation of an adaptive time steplength control. Consider a Runge–Kutta method of order $p \geq 2$. An adaptive time step control employs a second Runge–Kutta method which has the coefficients a_{ij}, $\hat{b}_i$ and c_i, $i, j = 1, \ldots, s$, and order $p-1$. The solution of the second method at time t_{m+1} is given by

$$\hat{\mathbf{u}}_{m+1} = \mathbf{u}_m + \sum_{i=1}^{s} \hat{b}_i \mathbf{k}_i.$$

Now, the next time step τ_{m+1} is proposed to be

$$\tau_{m+1} = \rho \frac{\tau_m^2}{\tau_{m-1}} \left(\frac{TOL \cdot r_m}{r_{m+1}^2} \right)^{1/p}, \tag{4}$$

where $\rho \in (0, 1]$ is a safety factor, $TOL > 0$ is a given tolerance and

$$r_{m+1} := \left\| \mathbf{u}_{m+1} - \hat{\mathbf{u}}_{m+1} \right\|. \tag{5}$$

This step size selection rule is called PI–controller and goes back to Gustafsson et al. [5]. For details on the numerical error and the implementation of automatic steplength control we refer to [7, 11].

2.2 Application to the example of Prothero–Robinson

In the following the consider the example of Prothero–Robinson, which is given by

$$\dot{u} = \lambda(u - \varphi(t)) + \dot{\varphi}(t), \quad u(0) = \varphi(0), \lambda \in \mathbb{R}^-, \tag{6}$$

where $\lambda \ll 0$ and $\varphi(t)$ is a given function. The exact solution of equation (6) is given by $u(t) = \varphi(t)$. Next we apply the ESDIRK–method (2)–(3) on the ODE (6). We obtain

$$k_i = \lambda \left(u_m + \tau \sum_{j=1}^{i} a_{ij} k_j - \varphi(t_m + c_i \tau) \right) + \dot{\varphi}(t_m + c_i \tau), \qquad i = 1, \ldots, s.$$

With the notations

$$\begin{aligned}
&\varphi_m^{(k)} := \varphi^{(k)}(t_m), \varphi_i^{(k)} := \varphi^{(k)}(t_m + c_i \tau), \quad i = 1, \ldots, s, k \geq 0, \\
&\tilde{\mathbf{\Phi}}^{(k)} := (\varphi_2^{(k)}, \ldots, \varphi_s^{(k)})^\top, \quad \tilde{\mathbf{k}} := (k_2, \ldots, k_s)^\top, \quad \tilde{\mathbf{e}} := (1, \ldots, 1)^\top \in \mathbb{R}^{s-1}, \\
&\quad \tilde{\mathbf{c}} := (c_2, \ldots, c_s)^\top
\end{aligned}$$

it follows

$$k_1 = \lambda(u_m - \varphi_m) + \dot{\varphi}_m,$$

$$k_i = \lambda\left(u_m + \tau\sum_{j=1}^{i} a_{ij}k_j - \varphi_i\right) + \dot{\varphi}_i$$

$$= \lambda\left(u_m + \tau a_{i1}k_1 + \tau\sum_{j=2}^{i} a_{ij}k_j - \varphi_i\right) + \dot{\varphi}_i, \quad i = 2, \ldots, s.$$

Using the vector notation introduced above we obtain

$$\tilde{\mathbf{k}} = \lambda(u_m\tilde{\mathbf{e}} + \tau\mathbf{a}_1 k_1 + \tau\tilde{A}\tilde{\mathbf{k}} - \Phi) + \dot{\Phi},$$

where

$$\mathbf{a}_1 = (a_{21}, \ldots, a_{s1})^\top, \quad \tilde{A} = \begin{pmatrix} a_{21} & \ldots & a_{2s} \\ \vdots & & \vdots \\ a_{s1} & \ldots & a_{ss} \end{pmatrix}.$$

With $z = \lambda\tau$ it follows

$$\tilde{\mathbf{k}} = (I - z\tilde{A})^{-1}(\lambda(u_m\tilde{\mathbf{e}} + \tau\mathbf{a}_1 k_1 - \tilde{\Phi}) + \dot{\tilde{\Phi}}). \tag{7}$$

Inserting into (7) into (3) yields

$$\begin{aligned} u_{m+1} &= u_m + \tau\sum_{i=1}^{s} b_i k_i = u_m + \tau b_1 k_1 + \tau\tilde{\mathbf{b}}^\top\tilde{\mathbf{k}} \\ &= u_m + \tau b_1 k_1 + \tau\tilde{\mathbf{b}}^\top(I - z\tilde{A})^{-1}[\lambda(u_m\tilde{\mathbf{e}} + \tau\mathbf{a}_1 k_1 - \tilde{\Phi}) + \dot{\tilde{\Phi}}] \\ &= u_m + \tau b_1 k_1 + \tau\tilde{\mathbf{b}}^\top(I - z\tilde{A})^{-1}\lambda(u_m\tilde{\mathbf{e}} + \tau\mathbf{a}_1 k_1 - \tilde{\Phi}) \\ &+ \tau\tilde{\mathbf{b}}^\top(I - z\tilde{A})^{-1}\dot{\tilde{\Phi}}, \end{aligned} \tag{8}$$

where $\tilde{\mathbf{b}} = (b_2, \ldots, b_s)^\top$.

2.3 New order conditions

Next we compute the local error of the ESDIRK–methods if they are applied to the Prothero–Robinson example. We have

$$\begin{aligned} \delta_\tau(t_{m+1}) &= u_{m+1} - \varphi(t_{m+1}) \\ &= \varphi_m - \varphi_{m+1} + \tau b_1 k_1 + z\tilde{\mathbf{b}}^\top(I - z\tilde{A})^{-1}[u_m\tilde{\mathbf{e}} + \tau\mathbf{a}_1 k_1 - \tilde{\Phi}] \\ &+ \tau\tilde{\mathbf{b}}^\top(I - z\tilde{A})^{-1}\dot{\tilde{\Phi}}]. \end{aligned} \tag{9}$$

In the non-stiff case we have $z \to 0$ for $\tau \to 0$, but in the stiff case z tends to infinity if $\tau \to 0$ [7]. We expand the term $(I - z\tilde{A})^{-1}$ for large z as follows

$$(I - z\tilde{A})^{-1} = -(\tilde{A}z)^{-1} - (\tilde{A}z)^{-2} - \ldots .$$

Since $(I - z\tilde{A})^{-1}$ is expanded in two variables τ and z we need the derivatives of $(I - z\tilde{A})^{-1}$. Let $\tilde{z} = 1/z$. Then

$$\left[\left(I - \frac{\tilde{A}}{\tilde{z}} \right)^{-1} \right]^{(k)} \to -\frac{k!}{\tilde{A}^k}, \qquad \text{for } \tilde{z} \to 0,$$

if $k \geq 1$. Then the Taylor expansion of $\delta_\tau(t_{m+1})$ reads as

$$\begin{aligned}
\delta_\tau(t_{m+1}) &= u_{m+1} - \varphi(t_{m+1}) \\
&= -\sum_{k=1}^{p} \varphi_m^{(k)} \frac{\tau^k}{k!} + \tau b_1 k_1 \\
&\quad - z \sum_{k=1}^{p+1} \tilde{\mathbf{b}}^\top \sum_{l=1}^{k} \binom{k}{l} l! \tilde{A}^{-l} [\varphi_m \tilde{\mathbf{e}} \delta_{k-l,0} + \mathbf{a}_1 k_1 \delta_{k-l,1} - \tilde{\mathbf{c}}^{k-l} \varphi_m^{(k-l)}] \frac{\tau^{k-l}}{k! z^l} \\
&\quad - \tau \sum_{k=1}^{p-1} \tilde{\mathbf{b}}^\top \sum_{l=1}^{k} \binom{k}{l} l! \tilde{A}^{-l} \tilde{\mathbf{c}}^{k-l} \varphi_m^{(k-l+1)} \frac{\tau^{k-l}}{k! z^l} + \mathcal{O}(\tau^{p+1}).
\end{aligned}$$

In the second row the term $k = l$ vanishes. In the last term we sum from $l = 0$ to $k - 1$. Since $u_m = \varphi_m$ it follows $k_1 = \dot{\varphi}_m$ and

$$\begin{aligned}
\delta_\tau(t_{m+1}) &= u_{m+1} - \varphi(t_{m+1}) \\
&= -\sum_{k=1}^{p} \varphi_m^{(k)} \frac{\tau^k}{k!} + \tau b_1 \dot{\varphi}_m \\
&\quad - \sum_{k=2}^{p+1} \tilde{\mathbf{b}}^\top \sum_{l=1}^{k-1} \tilde{A}^{-l} [\mathbf{a}_1 \dot{\varphi}_m \delta_{k-l,1} - \tilde{\mathbf{c}}^{k-l} \varphi_m^{(k-l)}] \frac{\tau^{k-l}}{(k-l)! z^{l-1}} \\
&\quad - \sum_{k=1}^{p-1} \tilde{\mathbf{b}}^\top \sum_{l=0}^{k-1} \tilde{A}^{-l} \tilde{\mathbf{c}}^{k-l-1} \varphi_m^{(k-l)} \frac{\tau^{k-l}}{(k-l-1)! z^{l+1}} + \mathcal{O}(\tau^{p+1}).
\end{aligned}$$

Then we split the second sum and obtain

$$\begin{aligned}\delta_\tau(t_{m+1}) = &-\sum_{k=1}^{p}\varphi_m^{(k)}\frac{\tau^k}{k!}+\mathcal{O}(\tau^{p+1})+\tau b_1\dot\varphi_m \\ &-\sum_{k=2}^{p+1}\tilde{\mathbf{b}}^\top\tilde A^{-1}[\mathbf{a}_1\dot\varphi_m\delta_{k-1,1}-\tilde{\mathbf{c}}^{k-1}\varphi_m^{(k-1)}]\frac{\tau^{k-1}}{(k-1)!} \\ &-\sum_{k=3}^{p+1}\tilde{\mathbf{b}}^\top\sum_{l=2}^{k-1}\tilde A^{-l}[\mathbf{a}_1\dot\varphi_m\delta_{k-l,1}-\tilde{\mathbf{c}}^{k-l}\varphi_m^{(k-l)}]\frac{\tau^{k-l}}{(k-l)!z^{l-1}} \\ &-\sum_{k=1}^{p-1}\tilde{\mathbf{b}}^\top\sum_{l=0}^{k-1}\tilde A^{-l-1}\tilde{\mathbf{c}}^{k-l-1}\varphi_m^{(k-l)}\frac{\tau^{k-l}}{(k-l-1)!z^{l+1}}.\end{aligned}$$

Next we sum in the second row from $k=1$ to p and combine this sum to the sum in the first row. Moreover we sum in the third and fourth row from $k=2$ to p. We obtain

$$\begin{aligned}\delta_\tau(t_{m+1}) &= u_{m+1}-\varphi(t_{m+1}) \\ &= \tau[b_1+\tilde{\mathbf{b}}^\top\tilde A^{-1}(\tilde{\mathbf{c}}-\mathbf{a}_1)-1]\dot\varphi_m \\ &+\sum_{k=2}^{p}\left[\tilde{\mathbf{b}}^\top\tilde A^{-1}\tilde{\mathbf{c}}^k-1\right]\varphi_m^{(k)}]\frac{\tau^k}{k!}+\mathcal{O}(\tau^{p+1}) \\ &-\sum_{k=2}^{p}\tilde{\mathbf{b}}^\top\sum_{l=1}^{k-1}\tilde A^{-l-1}[\mathbf{a}_1\dot\varphi_m\delta_{k-l,1}-\tilde{\mathbf{c}}^{k-l}\varphi_m^{(k-l)}]\frac{\tau^{k-l}}{(k-l)!z^l} \\ &-\sum_{k=2}^{p}\tilde{\mathbf{b}}^\top\sum_{l=1}^{k-1}\tilde A^{-l}\tilde{\mathbf{c}}^{k-l-1}\varphi_m^{(k-l)}\frac{\tau^{k-l}}{(k-l-1)!z^l}.\end{aligned}$$

The last two sums can be combined. Moreover we separate the case $l=k-1$. Then we have

$$\begin{aligned}\delta_\tau(t_{m+1}) = &\,\tau[b_1+\tilde{\mathbf{b}}^\top\tilde A^{-1}(\tilde{\mathbf{c}}-\mathbf{a}_1)-1]\dot\varphi_m \\ &+\sum_{k=2}^{p}\left[\tilde{\mathbf{b}}^\top\tilde A^{-1}\tilde{\mathbf{c}}^k-1\right]\varphi_m^{(k)}]\frac{\tau^k}{k!}+\mathcal{O}(\tau^{p+1}) \\ &+\sum_{k=2}^{p}\tilde{\mathbf{b}}^\top\left[\tilde A^{-k}(\tilde{\mathbf{c}}-\mathbf{a}_1)-\tilde A^{-k+1}\tilde{\mathbf{e}}\right]\dot\varphi_m\frac{\tau}{z^{k-1}} \\ &-\sum_{k=2}^{p}\tilde{\mathbf{b}}^\top\sum_{l=1}^{k-2}\tilde A^{-l}\left[\tilde A^{-1}\tilde{\mathbf{c}}^{k-l}-(k-l)\tilde{\mathbf{c}}^{k-l-1}\right]\varphi_m^{(k-l)}\frac{\tau^{k-l}}{(k-l)!z^l}.\end{aligned}$$

Finally we get the new order conditions

$$b_1 + \tilde{\mathbf{b}}^\top \tilde{A}^{-1}(\tilde{\mathbf{c}} - \mathbf{a}_1) = 1 \tag{10}$$

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-1}\tilde{\mathbf{c}}^k = 1, \quad k = 2, \dots, p, \tag{11}$$

$$\tilde{\mathbf{b}}^\top \left[\tilde{A}^{-k}(\tilde{\mathbf{c}} - \mathbf{a}_1) - \tilde{A}^{-k+1}\tilde{\mathbf{e}}\right] = 0, \quad k = 2, \dots, p, \tag{12}$$

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-l}\left[\tilde{A}^{-1}\tilde{\mathbf{c}}^{k-l} - (k-l)\tilde{\mathbf{c}}^{k-l-1}\right] = 0, \tag{13}$$

for $k = 2, \dots, p$ and $l = 1, \dots, k-2$.

Theorem 1.
- *If the ESDIRK method (2)–(3) is stiffly accurate the conditions (10) and (11) are automatically satisfied.*
- *Let the ESDIRK method (2)–(3) be consistent. Also, the simplifying condition $C(1)$ should be valid. Then condition (10) is fulfilled.*
- *If the ESDIRK method (2)–(3) satisfies the simplifying condition $C(2)$ equation (11) is automatically valid.*

Proof.
- First we consider the condition (10) and write it component-wise as

  $$b_1 + \sum_{i,j=2}^{s} \tilde{b}_i \omega_{ij}(c_j - a_{j1}) = 1,$$

  where the entries of $\tilde{A}^{-1}$ are denoted by ω_{ij}. Since $\tilde{b}_i = \tilde{a}_{si}$ holds for a stiffly accurate ESDIRK method we have

  $$\sum_{i=2}^{s} \tilde{b}_i \omega_{ij} = \sum_{i=2}^{s} \tilde{a}_{si}\omega_{ij} = \delta_{sj}$$

  and

  $$b_1 + \sum_{i,j=2}^{s} \tilde{b}_i \omega_{ij}(c_j - a_{j1}) = b_1 + \sum_{j=2}^{s} \delta_{sj}(c_j - a_{j1}) = b_1 + c_s - a_{s1} = 1.$$

  Equation (11) can be proven in an analogous way.

- The simplifying condition $C(1)$ can be written as

  $$\mathbf{a}_1 + \tilde{A}\mathbf{e} = \tilde{\mathbf{c}}. \tag{14}$$

  Next we insert equation (14) into condition (10) and obtain

  $$b_1 + \tilde{\mathbf{b}}^\top \tilde{A}^{-1}(\tilde{\mathbf{c}} - \mathbf{a}_1) = b_1 + \tilde{\mathbf{b}}^\top \tilde{A}^{-1}(\mathbf{a}_1 + \tilde{A}\mathbf{e} - \mathbf{a}_1) = b_1 + \tilde{\mathbf{b}}^\top \mathbf{e} = 1.$$

- The simplifying condition $C(2)$ can be written as $2\tilde{A}\tilde{\mathbf{c}} = \tilde{\mathbf{c}}^2$. It follows

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-1}\tilde{\mathbf{c}}^2 = 2\tilde{\mathbf{b}}^\top \tilde{A}^{-1}\tilde{A}\tilde{\mathbf{c}} = 2\tilde{\mathbf{b}}^\top \tilde{\mathbf{c}} = 1$$

$\square$

Theorem 2. *If the ESDIRK method (2)–(3) is consistent and satisfies the simplifying condition $C(1)$ condition (12) is automatically satisfied.*

Proof. Inserting the simplifying condition $C(1)$, i.e. equation (14), into (12) gives us

$$\tilde{\mathbf{b}}^\top \left[\tilde{A}^{-k}(\mathbf{a}_1 + \tilde{A}\mathbf{e} - \mathbf{a}_1) - \tilde{A}^{-k+1}\tilde{\mathbf{e}}\right] = 0$$

for all $k = 2, 3, \ldots$ and everything is proven. $\square$

Theorem 3. *If the ESDIRK method (2)–(3) satisfies the simplifying condition $C(2)$ condition (13) is satisfied for $l = k - 2$, $k = 3, 4, \ldots$.*

Proof. The simplifying condition $C(2)$ is given by $A\mathbf{c} = \mathbf{c}^2/2$ or componentwise by

$$\sum_{j=1}^{s} a_{ij}c_j = \frac{c_i^2}{2}.$$

It follows

$$\sum_{j=2}^{s} a_{ij}c_j = \frac{c_i^2}{2},$$

since $c_1 = 0$, and finally $\tilde{A}\tilde{\mathbf{c}} = \tilde{\mathbf{c}}^2/2$. For $l = k - 2$ the condition (13) is given by

$$\tilde{\mathbf{b}}^\top \tilde{A}^{2-k}\left[\tilde{A}^{-1}\tilde{\mathbf{c}}^2 - 2\tilde{\mathbf{c}}\right] = 0$$

Inserting $C(2)$ yields

$$\tilde{\mathbf{b}}^\top \tilde{A}^{2-k}\left[2\tilde{A}^{-1}\tilde{A}\tilde{\mathbf{c}} - 2\tilde{\mathbf{c}}\right] = 0$$

and everything is proven. $\square$

3 New ESDIRK methods

In this section we develop new stiffly accurate ESDIRK methods which satisfy the new order conditions (11) and (13). Moreover, adaptivity with an embedded method should be possible. Since Theorem 1 is valid for the embedded methods, too, all our embedded method will satisfy (10) and (11) for $k = 2$.

3.1 The ESDIRKPR53 method

First we create a stiffly accurate ESDIRK method of order 3 with 5 internal stages. The method should satisfy the simplifying conditions $B(1), B(2), B(3)$, $C(1)$ and $C(2)$. Also, condition (13) should be be valid for $k = 4$ and $l = 1$, $k = 5$ and $l = 2$. Moreover the method and its embedded method should be L-stable. If we use all the simple conditions such as $C(1)$ we have 13 degrees of freedom and 10 equations. The free coefficients are $c_2 = 5/9$, $c_4 = 9/10$ and $\hat{b}_4 = 1/2$. The remaining coefficients can be computed with the help of a computer algebra tool. The coefficients can be found in Table 3 in the Appendix.

3.2 The ESDIRKPR63 method

Next we want to improve the ESDIRKPR53 method in such a way that the embedded method is stiffly accurate, too. We need 6 internal stages and therefore we have more coefficients than in the previous case. We want to fulfill condition (13) for $k = 4$ and $l = 1$, $k = 5$ and $l = 2$, $k = 6$ and $l = 3$, $k = 5$ and $l = 1$. Then we have 11 equations and 13 variables. The variable c_2 is set to $5/6$ and c_4 to $3/10$. Again the coefficients are determined with the help of a computer algebra tool and are given in Table 4 in the Appendix.

3.3 The ESDIRKPR74 method

Next we want to find a 4th order ESDIRK method which satisfies the new order conditions. Therefore we need 7 internal stages. In this case the simplifying condition $B(4)$ should be fulfilled, too. Moreover condition (13) should be valid for $k = 4$ and $l = 1$, $k = 5$ and $l = 2$, $k = 6$ and $l = 3$, $k = 5$ and $l = 1$, $k = 6$ and $l = 2$. The free variables are chosen in the following way: $c_2 = 1/3$, $c_3 = 1/6$, $c_4 = 2/3$, $c_5 = 3/4$, $c_6 = 6/7$, $\hat{b}_2 = 1/10$, $\hat{b}_4 = 0$, and $a_{65} = 1/10$. Table 5 in the Appendix presents the coefficients of the method.

3.4 Comparison of methods

In this section we compare different ESDIRK methods. It is interesting to know which order conditions are satisfied by the different methods. In the literature many ESDIRK methods can be found. The following list of ESDIRK methods is of course not complete.

First we mention ESDIRK methods of order 3 and 4, which can be found in the paper of Kennedy and Carpenter [9]. These methods have 4 and 6 internal stages. In [10] several ESDIRK methods of order 3, 4 and 5 were created. In newer papers ESDIRK methods are developed which satisfy order conditions for DAEs of index 2 or higher. One example is in the paper of Williams et al. [19] with the method ESDIRK32 which has 4 internal stages and order 3. In [17] Skvortsov considers methods of order 4, which can be used for the solution of index-2 and index-3 DAEs.

Order conditions for index-2 DAEs were derived in the book of Hairer, Lubich and Roche [6]. Higueras simplifies these conditions (see [8]). The conditions are given by [19]

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-1}\tilde{\mathbf{c}}^k = 1, \qquad k \in \{1,2,3\}, \tag{15}$$

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-2}\tilde{\mathbf{c}}^k = k, \qquad k \in \{1,2,3\}. \tag{16}$$

Theorem 4. *Let a stiffly accurate ESDIRK method be given. If this method satisfies condition (13) for $l = 1$ and $k \in \mathbb{N}$ the index-2 conditions (15) and (16) are automatically fulfilled.*

Proof. As we have shown before condition (15) is fulfilled for all k if the ESDIRK method is stiffly accurate. Condition (13) reads in the case $l = 1$ and $k \in \mathbb{N}$ as

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-2}\tilde{\mathbf{c}}^{k-1} = \tilde{\mathbf{b}}^\top \tilde{A}^{-1}(k-1)\tilde{\mathbf{c}}^{k-2}, \qquad k = 2, 3, \dots .$$

Since our ESDIRK method is stiffly accurate, the right-hand side of the last equation is equal to $k-1$. Next we make an index shift from $k-1$ to k and get

$$\tilde{\mathbf{b}}^\top \tilde{A}^{-2}\tilde{\mathbf{c}}^k = k, \qquad k = 1, 2, \dots .$$

This is the index-2 condition (16). □

In Table 1 it is shown which order conditions are satisfied by the considered ESDIRK methods. Condition (13) with $k - l = 2$ is not considered since it is satisfied by the simplifying condition $C(2)$, which is fulfilled by all methods. The second column with $k = 4$ and $l = 1$ is a part of index-2 condition (16). It is satisfied by the index-2 methods, i.e. ESDIRK32 and the methods from Skvortsov. For the other columns one has to distinguish between third and fourth conditions. Most of the other conditions are only satisfied by our new methods. Only some conditions are fulfilled by the methods from Skvortsov. In the next chapter we show that it is important that all conditions be satisfied. Then it is possible to guarantee full order 3 or 4 in the case of the Prothero–Robinson example.

Table 1: Order conditions satisfied by the selected ESDIRK methods

Name	s	$k=4$ $l=1$	5 2	6 3	5 1	6 2	reference
ESDIRK32	4	x	-	-	-	-	[19]
ESDIRK3	4	-	-	-	-	-	[9]
ESDIRK32a	4	-	-	-	-	-	[10]
ESDIRKPR53	6	x	x	-	-	-	Section 3.1
ESDIRKPR63	6	x	x	x	x	-	Section 3.2
ESDIRK4	6	-	-	-	-	-	[9]
ESDIRK43a	5	-	-	-	-	-	[10]
ESDIRK43b	5	-	-	-	-	-	[10]
Skvortsov4-3	6	x	x	-	-	-	[17]
Skvortsov4-4	6	x	x	-	-	-	[17]
Skvortsov4-5	8	x	-	-	x	-	[17]
ESDIRKPR74	7	x	x	x	x	x	Section 3.3

4 Numerical examples

In this section we apply our new ESDIRK methods to several test examples. In Table 2 we summarise the properties of the ESDIRK methods which are considered. We first consider the example of Prothero and Robinson [14] to show that the new methods have full order. Then we consider a DAE of index 2. Finally we will see that our new methods can solve DAEs of index 2 effectively. Note that the methods from Skvortsov [17] are not equipped with an embedded method. Therefore we use these methods only in the cases, where we solve our problems with equidistant timesteps.

4.1 Example of Prothero–Robinson

First we consider the well-known example from Prothero and Robinson which is given by

$$\dot{u} = \lambda(u - \varphi(t)) + \dot{\varphi}(t), \quad u(0) = \varphi(0), \quad \lambda < 0. \tag{17}$$

The exact solution is given by $u(t) = \varphi(t)$, where the function $\varphi(t)$ is given by

$$\varphi(t) = \sin\left(\frac{\pi}{4} + t\right).$$

The ODE is solved (17) with equidistant step sizes $\tau = \frac{1}{10\cdot 2^k}$, $k = 0, \ldots, 5$

Table 2: Properties of the selected ESDIRK methods

Name	s	p	q	$\lvert R(\infty)\rvert$	$\lvert\tilde{R}(\infty)\rvert$	reference
ESDIRK32	4	2	2	0.33	1	[19]
ESDIRK3	4	3	2	0	0.07	[9]
ESDIRK32a	4	3	2	0	0.96	[10]
ESDIRKPR53	5	3	2	0	0	3.1
ESDIRKPR63	6	3	2	0	0	3.2
ESDIRK4	6	4	2	0	0.07	[9]
ESDIRK43a	5	3	2	0	0.55	[10]
ESDIRK43b	5	3	2	0.72	0	[10]
Skvortsov4-3	6	4	2	0	-	[17]
Skvortsov4-4	6	4	2	$L(89.95°)$	-	[17]
Skvortsov4-5	8	5	2	$L(87.7°)$	-	[17]
ESDIRK63PR	5	3	2	0	0	3
ESDIRK74PR	6	4	2	0	0	3

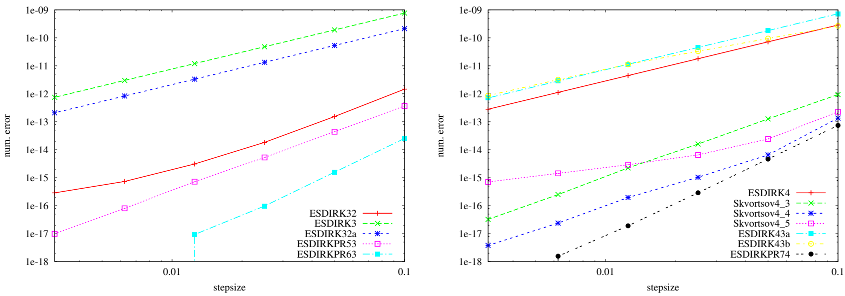

Figure 1: τ versus error for (17) with $\lambda = -10^6$: third order methods (left) and fourth order methods (right)

in the time interval $(0, 1/10]$. In Figure 1 we present the numerical results for $\lambda = -10^6$. In the left figure we compare the third order methods. It can be observed that the ESDIRK32 method converges with a higher order than usual third order methods which do not satisfy the index-2 conditions. The new methods ESDIRKPR53 and ESDIRKPR63 are the best methods in this case. The fourth order methods are compared in the right part of Figure 1.

Methods which are not designed for index-2 DAE converge with order 2 in the stiff case. An improvement can be observed if the methods are designed for index-2. The highest numerical order of convergence can be observed for our new method ESDIRKPR74.

4.2 An index-2 DAE

Next we consider the differential-algebraic equation (see [12], [4, Example 10] or [7, page 461]).

$$\left\{\begin{aligned} \dot{u}_1 - u_3\dot{u}_2 + u_2\dot{u}_3 &= 0 \\ u_2 &= \epsilon\sin(\omega t) \\ u_3 &= \epsilon\cos(\omega t) \\ u_1(0) &= 0 \end{aligned}\right. . \tag{18}$$

This problem has differentiation index 1, but perturbation index 2 ([7, page 461]). Numerical results applied on this problem should be designed in such a way, that order conditions for index-2 DAEs are satisfied (see [16]). The DAE (18) can be solved numerically by introducing new variables $z_i := \dot{u}_i$. With this setting it is possible to rewrite problem (18) in the form

$$\dot{\mathbf{u}} = \mathbf{z}, \qquad 0 = \mathbf{F}(t, \mathbf{u}, \mathbf{z}).$$

For our numerical experiments we chose $\epsilon = 1$ and $\omega = 25$. First we solve this problem in the time interval $[0, 1/10]$ with equidistant timesteps $\tau = 2.0 \cdot (1/2)^k$, where $k = 0, \ldots, 5$. We present the numerical results in Figure 2. We get similar results as in the case of the Prothero–Robinson

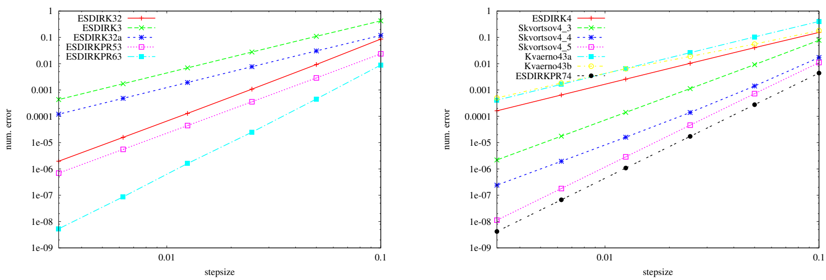

Figure 2: The solution of the index-2 DAE with constant stepsizes: third order methods (left) and fourth order methods (right)

example. Methods like ESDIRK32a, -43a and so on, which are not designed for index-2 DAEs, give the most inaccurate results. For the third order methods our new methods ESDIRKPR53 and ESDIRKPR63 are better than the ESDIRK32 method. In the case of the fourth order methods our new scheme ESDIRK74PR gives similar results as the Skvortsov4-5 method, but this method has no embedded method.

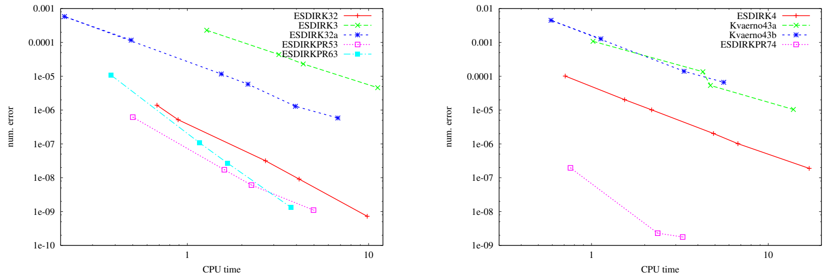

Figure 3: The solution of the index-2 DAE with adaptive timestep control: third order methods (left) and fourth order methods (right)

Next we solve the index-2 DAE (18) with an adaptive timestep control time interval $[0, 50]$, where we set $\omega = 10$. The numerical results are shown in Figure 3. In the left part we have the third order methods, which behave similar as in the previous simulation with constant stepsizes. Again we have the situation that the index-2 methods perform better than the others. Moreover, the new methods ESDIRKPR53 and ESDIRKPR63 are more effective than ESDIRK32. The same situation can be observed for the fourth order methods. Here the new ESDIRKPR74 method is the most effective one.

5 Summary and Outlook

In this paper we have considered ESDIRK methods and analysed the example of Prothero and Robinson. We have shown that further order conditions must be satisfied if we want to have full order convergence. Therefore we create new third and fourth order ESDIRK methods, which perform in our test examples much better than the usual ESDIRK methods.

References

[1] J. W. Butcher. On Runge–Kutta processes of high order. *J. Austral. Math. Soc.*, 4:179–194, 1964.

[2] F. Cameron. A class of low order DIRK methods for a class of DAEs. *Appl. Numer. Math.*, 31(1):1–16, 1999.

[3] Frank Cameron, Mikko Palmroth, and Robert Piché. Quasi stage order conditions for SDIRK methods. *Appl. Numer. Math.*, 42(1-3):61–75, 2002. doi:10.1016/S0168-9274(01)00142-8.

[4] S. Campbell and W. Gear. The index of general nonlinear DAEs. *Numer. Math.*, 72(2):173–196, 1995.

[5] K. Gustafsson, M. Lundh, and G. Söderlind. A PI stepsize control for the numerical solution of ordinary differential equations. *BIT*, 28(2):270–287, 1988.

[6] E. Hairer, C. Lubich, and M. Roche. *The numerical solution of differential-algebraic systems by Runge-Kutta methods.* Springer-Verlag, Berlin, 1989.

[7] E. Hairer and G. Wanner. *Solving ordinary differential equations. II: Stiff and differential-algebraic problems*, volume 14 of *Springer Series in Computational Mathematics.* Springer-Verlag, Berlin, 1996.

[8] I. Higueras. On simplifying assumptions of Runge-Kutta methods for index 2 differential algebraic problems. *Computing*, 54(2):185–190, 1995. doi:10.1007/BF02238131.

[9] Christopher A. Kennedy and Mark H. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Applied Numerical Mathematics*, 44:139–181, 2003.

[10] A. Kvaerno. Singly diagonally implicit Runge-Kutta methods with an explicit first stage. *BIT Numerical Mathematics*, 44(3):489–502, 2004.

[11] J. Lang. *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems*, volume 16 of *Lecture Notes in Computational Science and Engineering.* Springer-Verlag, Berlin, 2001.

[12] C. Lubich. Linearly implicit extrapolation methods for differential-algebraic systems. *Numer. Math.*, 55:197–211, 1989.

[13] A. Ostermann and M. Roche. Runge-Kutta methods for partial differential equations and fractional orders of convergence. *Math. Comput.*, 59(200):403–420, 1992. doi:10.2307/2153064.

[14] A. Prothero and A. Robinson. On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. *Math. Comp.*, 28:145–162, 1974.

[15] J. Rang. An analysis of the Prothero-Robinson example for constructing new DIRK and ROW methods. *Journal of Computational and Applied Mathematics*, 262:105–114, 2014. doi:http://dx.doi.org/10.1016/j.cam.2013.09.062.

[16] J. Rang and L. Angermann. Remarks on the differentiation index and on the perturbation index of non-linear DAEs. Mathematik-Bericht 2005/3, Institut für Mathematik, TU Clausthal, 2005.

[17] L.M. Skvortsov. Diagonally implicit Runge-Kutta methods for differential algebraic equations of indices two and three. *Comput. Math., Math. Phys.*, 50(6):993–1005, 2010. doi:10.1134/S0965542510060072.

[18] K. Strehmel and R. Weiner. *Linear-implizite Runge–Kutta-Methoden und ihre Anwendung*, volume 127 of *Teubner-Texte zur Mathematik*. Teubner, Stuttgart, 1992.

[19] R. Williams, K. Burrage, I. Cameron, and M. Kerr. A four-stage index 2 diagonally implicit Runge-Kutta method. *Appl. Numer. Math.*, 40(3):415–432, 2002. doi:10.1016/S0168-9274(01)00090-3.

A Appendix: Coefficients of methods

a_{21}	=	$2.777777777777778e-01$	a_{22}	=	$2.777777777777778e-01$
a_{31}	=	$3.456552483519272e-01$	a_{32}	=	$1.681740315717733e-01$
a_{33}	=	$2.777777777777778e-01$	a_{41}	=	$3.965643047257401e-01$
a_{42}	=	$1.001154404932533e-01$	a_{43}	=	$1.255424770032288e-01$
a_{44}	=	$2.777777777777778e-01$	a_{51}	=	$2.481479828780141e-01$
a_{52}	=	$2.139473588935955e-01$	a_{53}	=	$1.206274239267400e+00$
a_{54}	=	$-9.461473588167871e-01$	a_{55}	=	$2.777777777777778e-01$
b_1	=	$2.481479828780141e-01$	$\hat{b}_1$	=	$4.445537532713554e-01$
b_2	=	$2.139473588935955e-01$	$\hat{b}_2$	=	$-1.065203443758999e-01$
b_3	=	$1.206274239267400e+00$	$\hat{b}_3$	=	$2.533129069755295e-01$
b_4	=	$-9.461473588167871e-01$	$\hat{b}_4$	=	$5.000000000000000e-01$
b_5	=	$2.777777777777778e-01$	$\hat{b}_5$	=	$-9.134631587098500e-02$

Table 3: Set of coefficients for the ESDIRKPR53 method

a_{21}	=	$4.166666666666667e-01$	a_{22}	=	$4.166666666666667e-01$
a_{31}	=	$3.640473915723038e-01$	a_{32}	=	$-4.189886135331312e-02$
a_{33}	=	$4.166666666666667e-01$	a_{41}	=	$-2.894969214392781e+00$
a_{42}	=	$-2.256341718064659e+01$	a_{43}	=	$2.534171972837271e+01$
a_{44}	=	$4.166666666666667e-01$	a_{51}	=	$2.309551022782098e-01$
a_{52}	=	$-1.849667242832423e+00$	a_{53}	=	$2.197073089164931e+00$
a_{54}	=	$4.972384722615363e-03$	a_{55}	=	$4.166666666666667e-01$
a_{61}	=	$3.054968378466108e-01$	a_{62}	=	$4.057983152922798e+00$
a_{63}	=	$-2.202162095667910e+00$	a_{64}	=	$1.333484429273537e-01$
a_{65}	=	$-1.711333004695519e+00$	a_{66}	=	$4.166666666666667e-01$
b_1	=	$3.054968378466108e-01$	$\hat{b}_1$	=	$2.309551022782098e-01$
b_2	=	$4.057983152922798e+00$	$\hat{b}_2$	=	$-1.849667242832423e+00$
b_3	=	$-2.202162095667910e+00$	$\hat{b}_3$	=	$2.197073089164931e+00$
b_4	=	$1.333484429273537e-01$	$\hat{b}_4$	=	$4.972384722615363e-03$
b_5	=	$-1.711333004695519e+00$	$\hat{b}_5$	=	$4.166666666666667e-01$
b_6	=	$4.166666666666667e-01$	$\hat{b}_6$	=	$0.000000000000000e+00$

Table 4: Set of coefficients for the ESDIRKPR63 method

a_{21}	$=$	$1.666666666666667e-01$	a_{22}	$=$	$1.666666666666667e-01$
a_{31}	$=$	$4.166666666666666e-02$	a_{32}	$=$	$-4.166666666666666e-02$
a_{33}	$=$	$1.666666666666667e-01$	a_{41}	$=$	$-1.500000000000000e+00$
a_{42}	$=$	$-1.333333333333333e+00$	a_{43}	$=$	$3.333333333333333e+00$
a_{44}	$=$	$1.666666666666667e-01$	a_{51}	$=$	$-1.580729166666667e+00$
a_{52}	$=$	$-1.349609375000000e+00$	a_{53}	$=$	$3.472656250000000e+00$
a_{54}	$=$	$4.101562500000000e-02$	a_{55}	$=$	$1.666666666666667e-01$
a_{61}	$=$	$-2.005366150605651e+00$	a_{62}	$=$	$-1.768688648609954e+00$
a_{63}	$=$	$4.341269295345690e+00$	a_{64}	$=$	$2.326169434610579e-02$
a_{65}	$=$	$1.000000000000000e-01$	a_{66}	$=$	$1.666666666666667e-01$
a_{71}	$=$	$1.684854267805816e-01$	a_{72}	$=$	$7.501080898831836e-01$
a_{73}	$=$	$-2.255843889686931e-01$	a_{74}	$=$	$-9.134421504267402e-01$
a_{75}	$=$	$1.618140253772232e+00$	a_{76}	$=$	$-5.643738977072310e-01$
a_{77}	$=$	$1.666666666666667e-01$			
b_1	$=$	$1.684854267805816e-01$	$\hat{b}_1$	$=$	$-3.930182461751728e-01$
b_2	$=$	$7.501080898831836e-01$	$\hat{b}_2$	$=$	$1.000000000000000e-01$
b_3	$=$	$-2.255843889686931e-01$	$\hat{b}_3$	$=$	$9.916346405575472e-01$
b_4	$=$	$-9.134421504267402e-01$	$\hat{b}_4$	$=$	$0.000000000000000e+00$
b_5	$=$	$1.618140253772232e+00$	$\hat{b}_5$	$=$	$-2.511232158528943e-01$
b_6	$=$	$-5.643738977072310e-01$	$\hat{b}_6$	$=$	$4.393912810497486e-01$
b_7	$=$	$1.666666666666667e-01$	$\hat{b}_7$	$=$	$1.131155404207712e-01$

Table 5: Set of coefficients for the ESDIRKPR74 method

Technische Universität Braunschweig

Informatik-Berichte ab Nr. 2011-10

2011-10	B. V. Rosić, A. Kučerová, J. Sýkora, A. Litvinenko, O. Pajonk and H. G. Matthies	Parameter Identification in a Probabilistic Setting
2011-11	M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies and E. Zander	Efficient Analysis of High Dimensional Data in Tensor Formats
2011-12	S. Oster	A Semantic Preserving Feature Model to CSP Transformation
2012-01	O. Pajonk, B. V. Rosić and H. G. Matthies	Deterministic Linear Bayesian Updating of State and Model Parameters for a Chaotic Model
2012-02	B. V. Rosić and H. G. Matthies	Stochasticc Plasticity - A Variational Inequality Formulation and Functional Approximation Approach I: The Linear Case
2012-03	J. Rang	An analysis of the Prothero–Robinson example for constructing new DIRK and ROW methods
2012-04	S. Kolatzki, M. Hagner, U. Goltz and A. Rausch	A Formal Definition for the Description of Distributed Concurrent Components - Extended Version
2012-05	M. Espig, W. Hackbusch, A. Litvinenko, H. G. Matthies and P. Wähnert	Efficient low-rank approximation of the stochastic Galerkin matrix in tensor formats
2012-06	S. Mennike	A Petri Net Semantics for the Join-Calculus

2012-07	S. Lity, R. Lachmann, M. Lochau, I. Schaefer	Delta-oriented Software Product Line Test Models - The Body Comfort System Case Study
2013-01	M. Lochau, S. Mennicke, J. Schroeter und T. Winkelmann	Extended Version of 'Automated Verification of Feature Model Configuration Processes based on Workflow Petri Nets'
2013-02	S. Lity, M. Lochau, U. Goltz	A Formal Operational Semantics of Sequential Function Tables for Model-based SPL Conformance Testing
2013-03	L. Giraldi, A. Litvinenko, D. Liu, H. G. Matthies, A. Nouy	To be or not to be intrusive? The solution of parametric and stochastic equations – the "plain vanilla" Galerkin case
2013-04	A. Litvinenko, H. G. Matthies	Inverse problems and uncertainty quantification
2013-05	J. Rang	Improved traditional Rosenbrock–Wanner methods for stiff ODEs and DAEs
2013-06	J. Koslowski	Deterministic single-state 2PDAs are Turing-complete
2014-01	B. Rosić, J. Diekmann	Stochastic Description of Aircraft Simulation Models and Numerical Approaches
2014-02	M. Krosche, W. Heinze	A Robustness Analysis of a Preliminary Design of a CESTOL Aircraft
2014-03	J. Rang	Apdative timestep control for fully implicit Runge–Kutta methods of higher order
2014-04	S. Mennicke, J.-W. Schicke-Uffmann, U. Goltz	Free-Choice Petri Nets and Step Branching Time
2014-05	A. Martens, C. Borchert, T. O. Geissler, O. Spinzcyk, D. Lohmann, R. Kapitza	Exploiting determinism for efficient protection against arbitrary state corruptions

2014-06	J. Rang	An analysis of the Prothero–Robinson example for constructing new adaptive ESDIRK methods of order 3 and 4